

Data WareHouse Architecture

Scalable Data Engineering

Motivation for a DWH

Preparation

- Multitude of different data sources
- Data cleansing / cleaning → historic reason for DWH

Adaptation/Standardization/Interpretability

- Central defined key performance indicators (KPI Management) and dimensions
- One single truth, no different interpretations

Allows new business processes

- Statistical methods (e.g. correlation analysis)
- Examples: customer segmentation, -evaluation

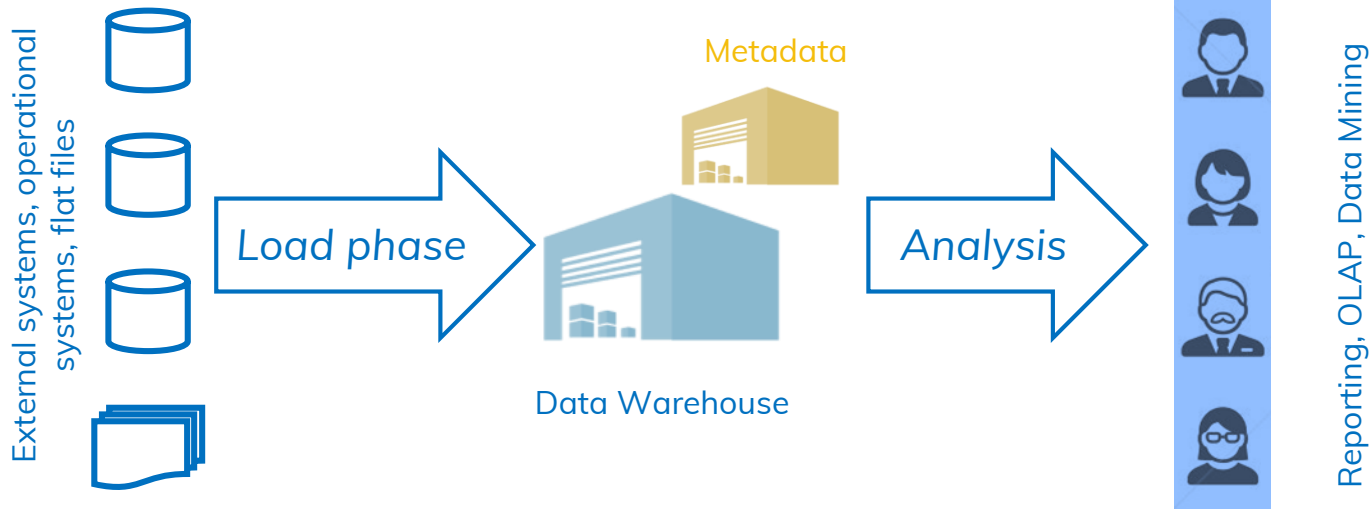
More

- Organization: data from multiple business units
- Technical: different query models, usage patterns, data volumes
- Consolidation: heterogeneity (schema, data quality, "garbage in, garbage out!")

Data-Warehouse concept

Goal: (analytic) access to consistent data

- Integration of external/operational data sources in a centralized database
- Extraction and multi-stage loading process
- Integrated (and historicized) data form a base for analytic tools



Descriptive Statistics

Processing step	Actions	Data
<i>Data recording</i>	Creation of raw data	Raw data „Microcosmos“
	calibration / checking	
	optional: anonymization	
<i>Data preparation</i>	Structural adaptation	processed raw data
	Validation (filtering, quality checking)	
	Statistical corrections (imputation, estimation, „missing values“)	
	Optional: anonymization, pre-aggregation	
<i>Data analysis</i>	Application-specific aggregation	Enriched data/aggregates, „Macrocosmos“
	Representation / interpretation	

Definition DWH

A data warehouse is a central database which is optimized for analysis purposes and contains data of several, usually heterogeneous data sources.

(Erhard Rahm)

Some general approaches

- *"Data warehouse is a **subject-oriented, integrated, time-varying, non-volatile** collection of data in support of the management's decision-making process."* (Bill Inmon)
- *"Data Warehouse is **an environment, not a product.**"* (Berson/ Smith)
- *"Physical database as an **integrated view** on (arbitrary) data with a **focus on the evaluation aspect**, where **data is often but not necessarily historized.**"* (GI)

Characteristics of DWHs

Analysis-oriented organization of data

- Domain-oriented, models a specific application goal

Integration of data from different source systems

- Integrated database, integration on structural and data level of multiple databases

No user updates

- Operations in operational environment
 - Insert, Delete, Update, Select
- Operations in a data warehouse
 - Insert: the initial and additional loading of data by (batch) processes
 - Select: the access of data
 - (almost) **no updates and deletes** (technical updates / deletes only, quality assurance)
- Non-volatile/stable database, loaded data is not deleted or modified, read-only access

Historic data

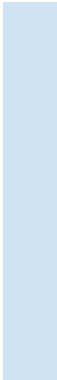
- Data is kept over a long period of time

What happens in the OLTP system if the customer cancels his booking?

- Delete operation in OLTP
- Seat gets available again and can be sold to another passenger

What happens in the DWH?

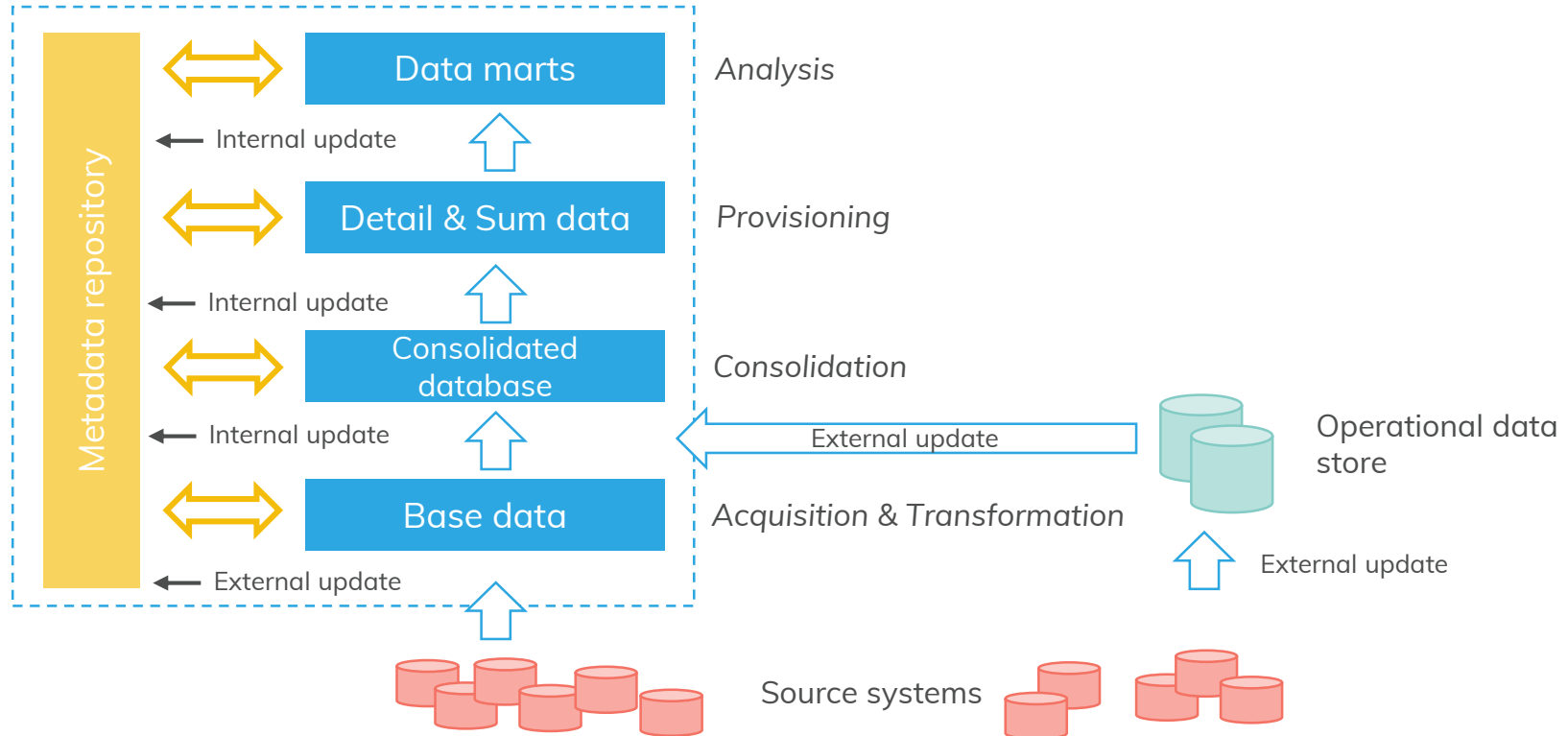
- Insert operation in DWH with, e.g., a flag indicating that the customer cancelled/deleted his booking
- Business can make analysis about cancelled booking: why might the customer have cancelled? How to prevent the customer or other customers to cancel next time?



DWH architecture

Reference architecture for DWH

Data-Warehouse-System



Example DWH architecture

Sources

Name	Gehalt	Name	Salary
Max Müller	3.500€	Tim Mueller	7,400\$

Example DWH architecture

Data transformation
Sources

Name	Salary
Max Mueller	3.500€
Tim Mueller	5.176€

Example DWH architecture

Data consolidation



Data transformation

Sources

Zeit	Name	Salary
T1	Max Mueller	3.300€
T1	Tim Mueller	6.176€
T2	Max Mueller	3.500€
T2	Tim Mueller	5.176€

Name	Salary
Max Mueller	3.500€
Tim Mueller	5.176€

Example DWH architecture

Data provisioning



Data consolidation



Data transformation

Sources

ZID	Time	ZID	MID	Salary	MID	Name
1	T1	1	1	3.300€	1	Max Mueller
2	T2	1	2	6.176€	2	Tim Mueller
		2	1	3.500€		
		2	2	5.176€		

Zeit	Name	Salary
T1	Max Mueller	3.300€
T1	Tim Mueller	6.176€
T2	Max Mueller	3.500€
T2	Tim Mueller	5.176€

Name	Salary
Max Mueller	3.500€
Tim Mueller	5.176€

Example DWH architecture

Data analytics



Data provisioning



Data consolidation



Data transformation

Sources

ZID	MID	Salary	ZID	MID	Salary
1	1	3.300€	1	2	6.176€
2	1	3.500€	2	2	5.176€



ZID	Time	ZID	MID	Salary	MID	Name
1	T1	1	1	3.300€	1	Max Mueller
2	T2	1	2	6.176€	2	Tim Mueller
		2	1	3.500€		
		2	2	5.176€		

Zeit	Name	Salary
T1	Max Mueller	3.300€
T1	Tim Mueller	6.176€
T2	Max Mueller	3.500€
T2	Tim Mueller	5.176€

Name	Salary
Max Mueller	3.500€
Tim Mueller	5.176€

What are data sources?

- Heterogeneous systems or files
- Local schemata and semantics

Describing attributes of data sources

- Utilization of data for the Data Warehouse
- Origin (internal or external data)
- Cooperation (active sources, snapshot-sources,)
- Availability of source data (legal, social, organizational, technical)
- Cost of acquisition
- Quality (consistency, correctness, completeness, accuracy,...)

Data acquisition and transformation

Staging Area

- “Landing Zone” for data coming into a DWH
- Temporary memory for integrating extracted data after an external update

Schema integration

- Overcoming semantic/structural heterogeneity
- Integration of different local schemata /data models into one global schema
- Decoupling of transformation from the source systems and the consolidated database

Data integration

- Adaptation of data formats etc.
- Correction of different spellings (abbreviations, etc.)

ETL-components

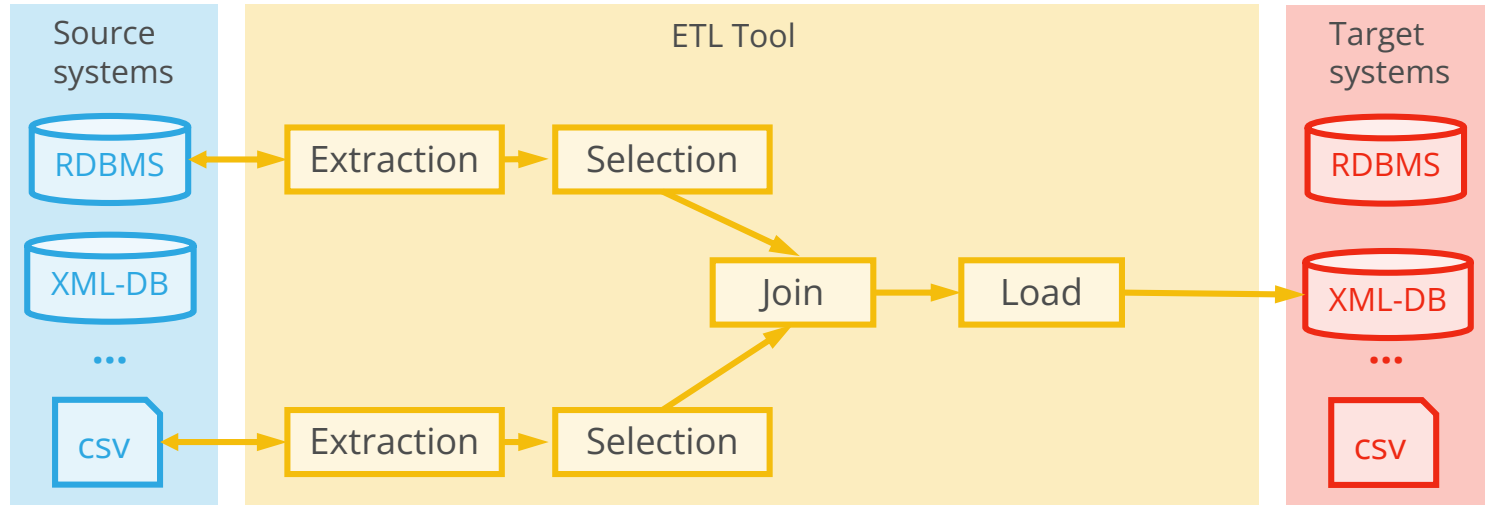
- Extraction-, Transformation- and Load component

Data acquisition and transformation

Goal

- Provisioning of data for the consolidated database
- Efficiency vs. Actuality

Architecture



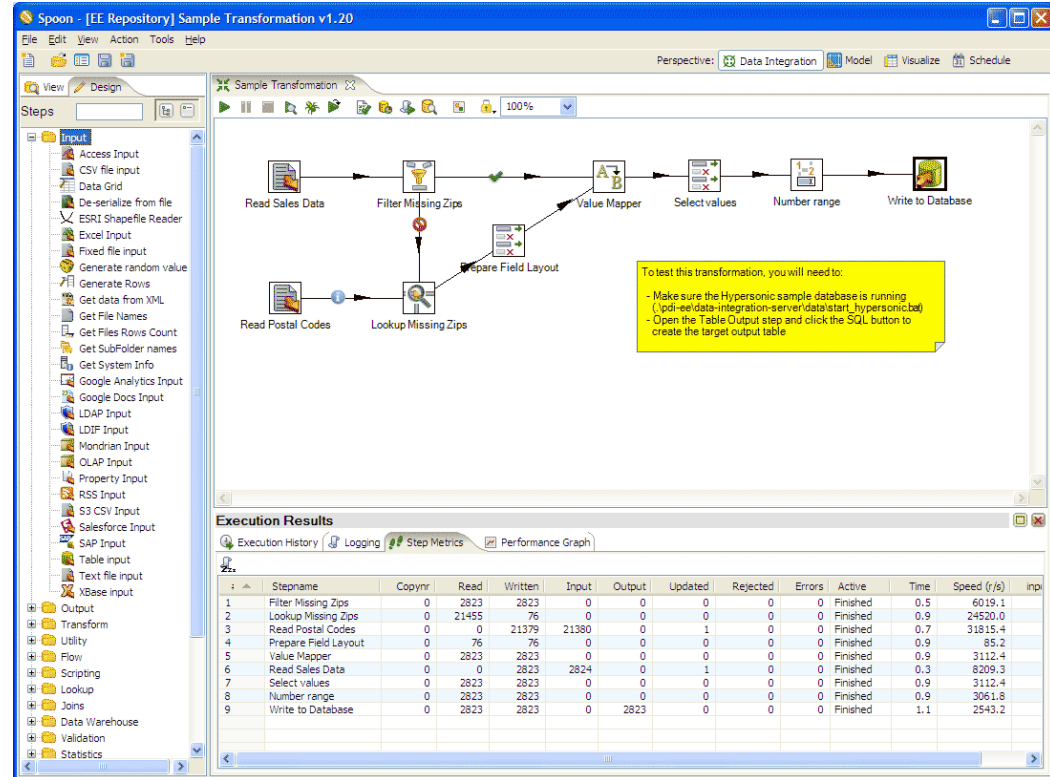
Example ETL Tools

Commercial Systems

- Informatica PowerCenter
- Cognos Decision-Stream
- Oracle DW Builder
- IBM InfoSphere DataStage
- IBM DB2 Warehouse Enterprise Edition
- AB Initio

Open Source

- Pentaho Data Integration (a.k.a Kettle)
- Talend ETL Integration Suite
- Clover ETL Data Integration
- JasperETL Open Source



Characteristics

- Integrated database from cleaned data
- Finest resolution / lowest granularity
- Not specifically modeled, no specific optimizations
 - For schema (classical logical and physical database design)
 - For physical database design (index structures, etc.)

Function

- Organization-spanning and application-independent data storage
- Collection, integration and distribution
- Analytic functionality for operative use

Update Alternatives

- Real-time
- Periodically
- Depending on the quantity of updates

Data provisioning and analysis

Data provisioning

- Dispositive database derived from consolidated data
- Mostly historic, maybe detailed data, mostly complete

Optimization for analysis

- Schema level: Star-/Snowflake-schema
- Logical access paths
- Partitioning
- Pre-calculation of summed data (e.g. materialized views)
- Physical access paths: specific index structures (e.g. bitmap index)

Data analysis

- Data-Mart databases derived from dispositive database
- Specific extracts for a specific class of applications
- Mostly proprietary formats on the physical level (e.g. MOLAP-systems)

Data-Mart concept

What?

- Structural/ content related subset of the complete DWH data

Why?

- Read-optimized layer: Data is stored in a denormalized data model for better read performance and better end user usability/understanding
- The Data Mart Layer is providing typically aggregated data or data with less history (e.g. latest years only) in a denormalized data model
- Dividing independent topics, ideally one subject per data mart
- Privacy aspects
- Reduction of data volume
- Performance/Load distribution

How?

- Geographic/Organization/Function/Competition oriented Data Marts
- Dependent (redundant) and independent Data Marts



Metadata

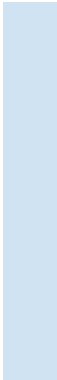
- Information concerning structure, maintenance, and administration of a Data-Warehouse-System, i.e., schema data and additional information
- Descriptive and process-related information

Examples

- Description of the Data Warehouse System
- Names, definitions, structure, and content of a DWH
- Identification of data sources
- Integration and transformation rules for filling the DWH
- Integration and transformation rules for end-user rules
- Operational information like updates, versions
- Usage and performance of the Data Warehouse (Monitoring)
- Security, access rights

Metadata





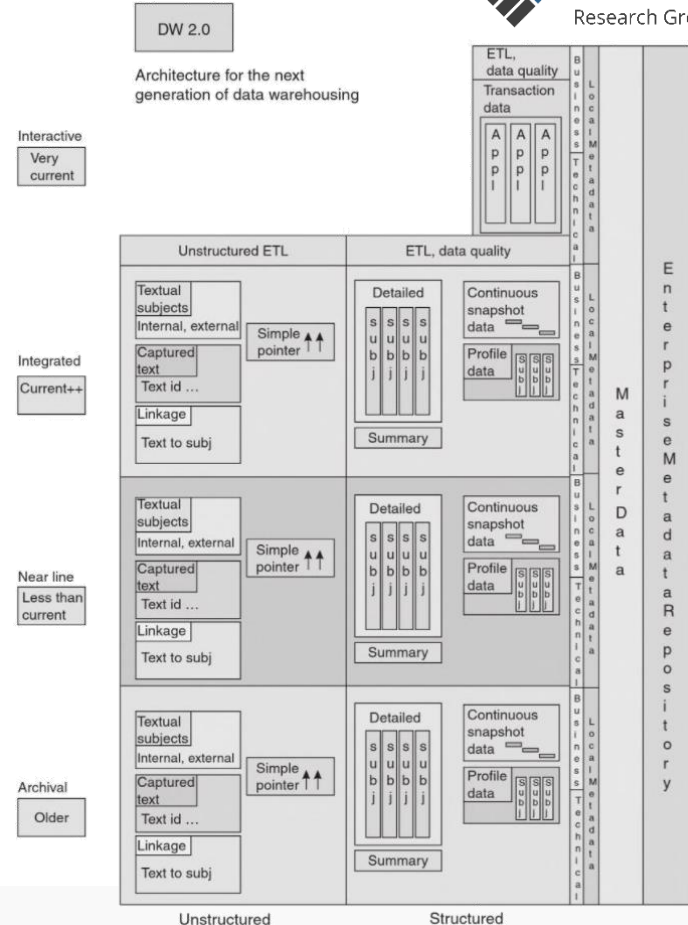
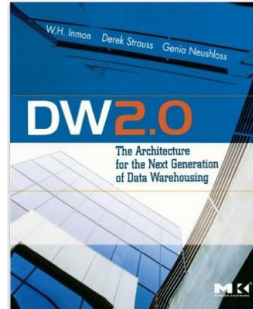
DWH Developments and Alternatives

DW 2.0: The Architecture for the Next Generation DWH

Main characteristics:

- Structured and “unstructured” data, not just metrics
- Life Cycle of data with different storage areas
 - Hot data: High speed, expensive storage (RAM, SSDs) for most recent data
 - Cold data: Low speed, inexpensive storage (e.g. hard disks) for old data; archival data model with high compression
- Metadata is an integral part of the DWH and not an afterthought

W.H. H. Inmon, Derek Strauss, Genia Neushloss: DW 2.0: The Architecture for the Next Generation of Data Warehousing



From Big Data to the Data Lake

With cheap storage costs, people promote the concept of the data lake

- Combines data from many sources and of any type
- Allows for conducting future analysis and not miss any opportunity

Collect everything

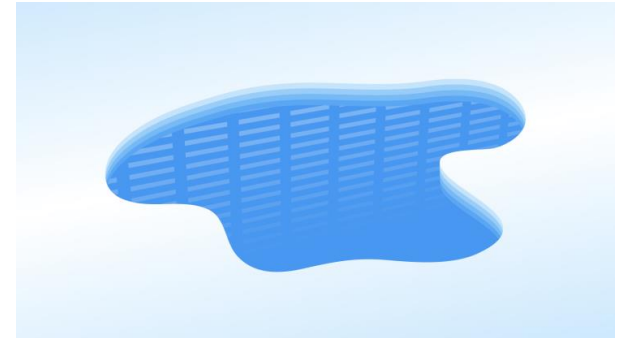
- All data, both raw sources over extended periods of time as well as any processed data
- Decide during analysis which data is important, e.g., no “schema” until read

Dive in anywhere

- Enable users across multiple business units to refine, explore and enrich data on their terms

Flexible access

- Enable multiple data access patterns across a shared infrastructure: batch, interactive, online, search, and others



Lambda Architecture (Big Data)

Batch Layer

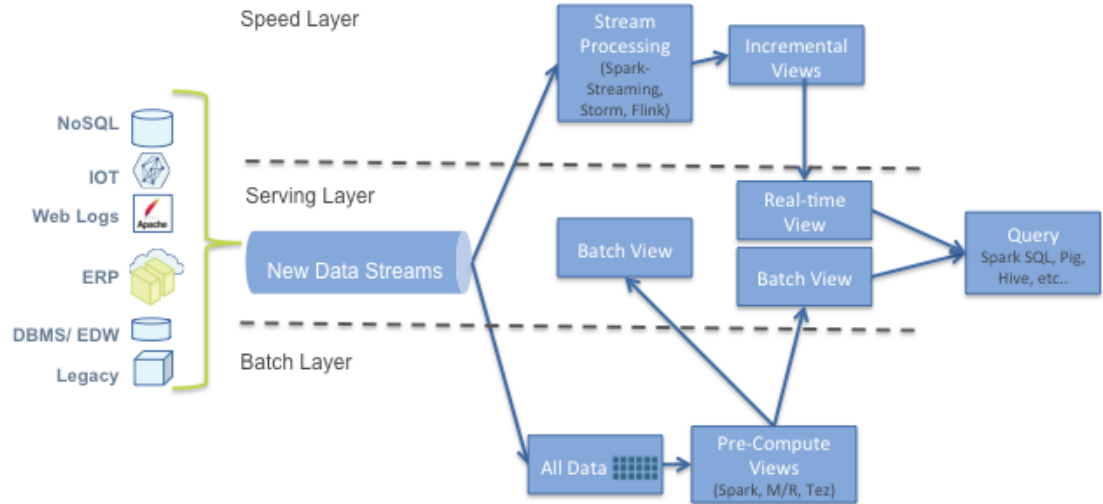
- DWH alike, correct & complete
- Output in Read-only DB, complete replacement
- Hadoop/Spark de facto standard

Speed Layer

- Stream processing
- Real-time, latency is king, „batch gap“
- Correct-/completeness minor
- Apache Storm, Spark etc.
- Output into fast NoSQL DBs

Serving Layer

- Query processing
- Stores outputs, builds views
- Druid, Cassandra, HBase



Logical Data Warehouse

Focus on Jobs to be done

DWH

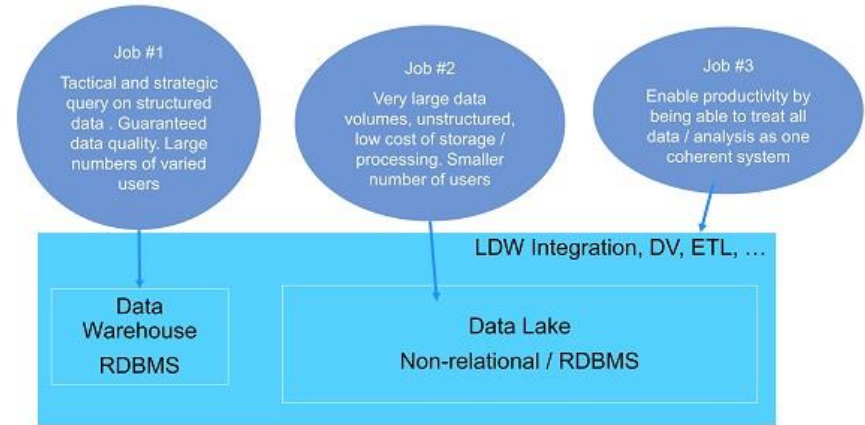
- Query processing on structured data
- Guaranteed quality

Data Lake

- Very large volumes of unstructured data

Logical Data Warehouse

- Provide access to both underlying systems
- Requires ETL functionality
- Virtual integration



[<https://blogs.gartner.com/henry-cook/2018/01/28/the-logical-data-warehouse-and-its-jobs-to-be-done/>]